National Aeronautics and Space Administration

Programming Challenges: NASA Advanced Supercomputing (NAS) Facility

Piyush Mehrotra, Henry Jin

NASA Advanced Supercomputing (NAS) Division

NASA Ames Research Center, Moffett Field, Calif., USA

March 2022

Supercomputing Center@ NASA Ames

AITKEN

Vital Stats

2,176-node HPE E-Cell/Apollo 9000 system

177,152 cores total

8.41 petaflops theoretical peak

5.79 petaflops sustained performance (June 2021)

745 terabytes total memory

ELECTRA

Vital Stats

3,456-node SGI/HPE ICE X/HPE E-Cell system

124,416 cores total

8.32 petaflops theoretical peak

5.44 petaflops sustained performance (June 2021)

589 terabytes total memory

PLEIADES

Vital Stats

11,207-node SGI/HPE ICE supercluster

241,324 cores total

7.09 petaflops theoretical peak

5.95 petaflops sustained performance (June 2021)

927 terabytes total memory

VISUALIZATION

Vital Stats

128-screen tiled LCD wall arranged in 8x16 configuration (23-ft. wide by 10-ft. high)

2,560 Intel Xeon Ivy Bridge processor cores

128 Nvidia GeForce GTX 780 Ti graphics processing units

NASA's Premier Supercomputer Center

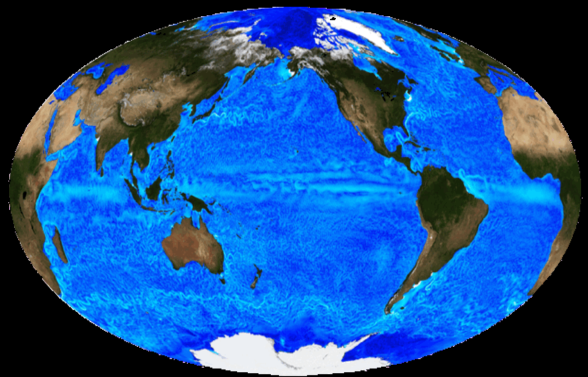
Resources have broad mission impact across all of NASA's Missions

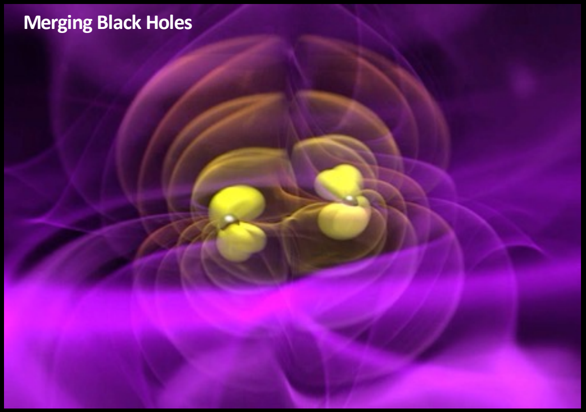
Over 600 science & engineering projects with more than 1,600 users

Example Computational Domains:
CFD for Vehicle design and analysis, materials, weather and climate modeling, oceanography, cosmology, exoplanet search, magneto-hydro-dynamics, space weather.

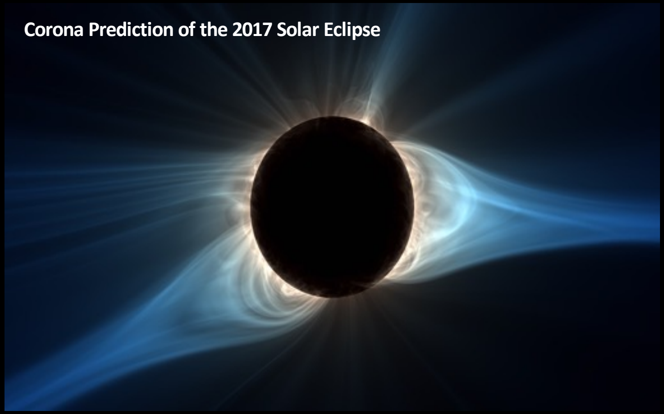
Representative applications @ NAS

ECCO: Global Ocean State



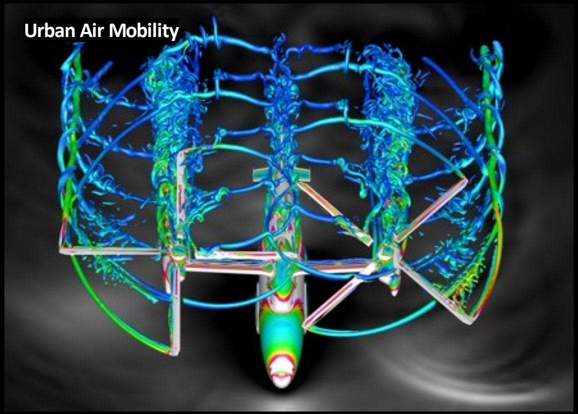


Merging Black Holes

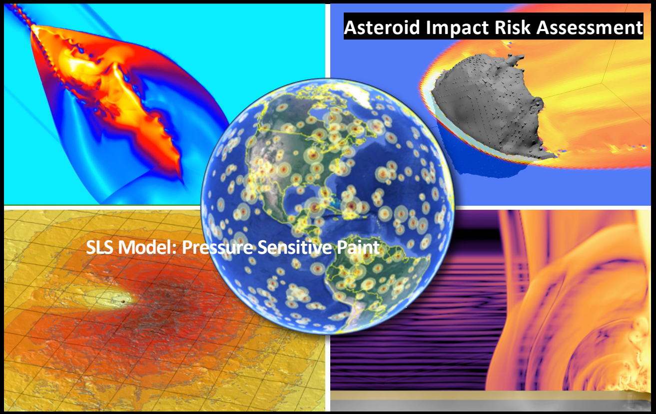


Corona Prediction of the 2017 Solar Eclipse

Landing Gear Noise

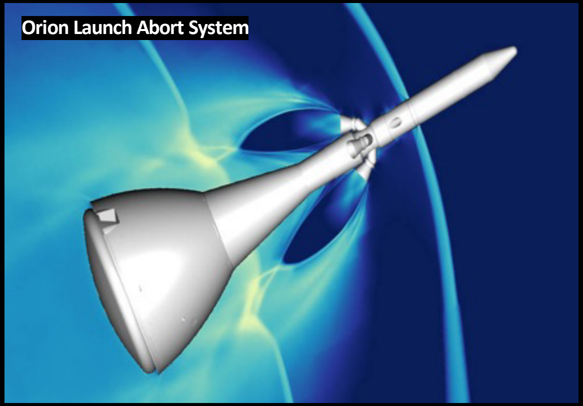


Urban Air Mobility



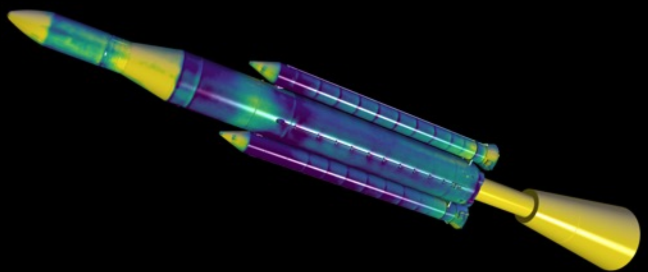
Asteroid Impact Risk Assessment

SLS Model: Pressure Sensitive Paint



Orion Launch Abort System

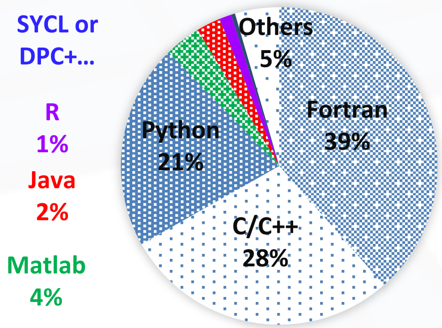
Launch Environment



Connecting the wind-tunnel to HPC

Programming Languages, Libraries, Commercial Software (2020 User Survey)

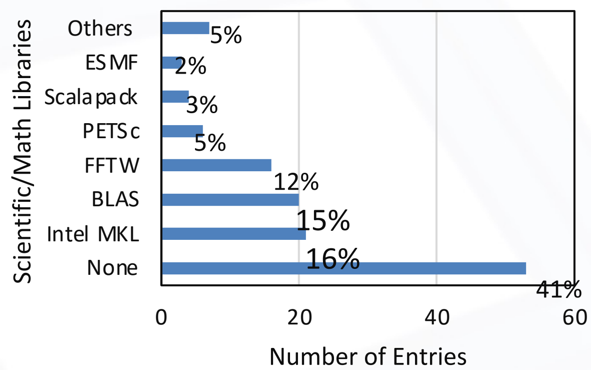
Programming Languages
(244 entries)



Others:
- Ruby (3 entries)
- Julia (2)
- CUDA/OpenMP (1)
- IDL (1)
- Tcl/tk (1)
- Shell scripting (1)
- Don’t know (2)

➢ Fortran/C/C++ still dominate.
➢ Python is getting popular.
➢ SYCL/DPC++ is being explored (by FUN3D developers).

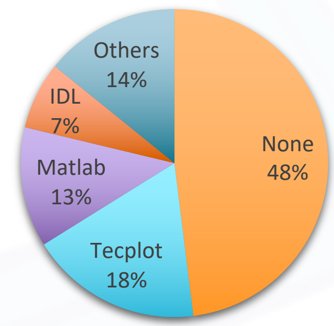
Scientific/Math Libraries
(130 entries)



Others:
- Armadillo (1)
- HYPRE, SLUG (1)
- Intel C runtime (1)
- Python (1)
- Don’t know (3)

➢ 59% of entries use sci/math libraries.
➢ Intel MKL, BLAS, FFTW dominate.

Commercial Software
(127 entries)



Other commercial: (8)
- Paraview (2)
- Powerflow (2)
- ANSA (1)
- CAMRADII (1)
- Pointwise (1)
- Totalview (1)

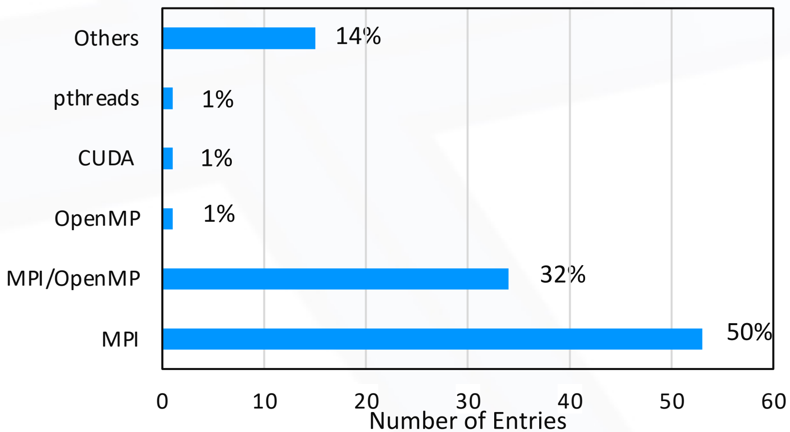
Non-commercial listed: (6)
- FITS, git, miniconda, netcdf,
- Python (2), tensorflow

Don’t know: (3)

➢ Licensed Tecplot/Matlab/IDL still in need.
➢ Open source software packages are popular.

Parallelism in Applications (2020 User Survey)

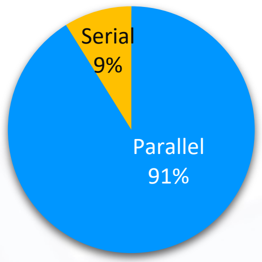
Parallel Paradigm
(105 entries)



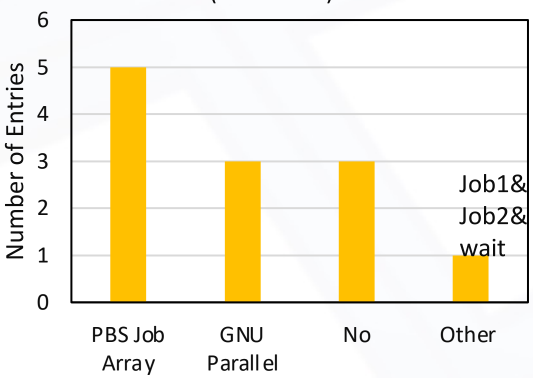
Others:
- Combination of
  - MPI/CUDA
  - MPI/OpenMP
  - MPI/OpenACC
  - MPI/pthreads
- SYCL
- Linda
- GNU Parallel (w/o MPI)
- OpenMP/Python multiprocessing

- ➢ MPI still dominates (~ 82% MPI or MPI/OpenMP).
- ➢ Pure OpenMP or pthreads not heavily used.
- ➢ CUDA programming begins to show up at HECC.
- ➢ Some interests in different hybrid parallelism: especially, MPI or MPI/OpenMP on CPU and MPI/CUDA on GPU.

Serial or Parallel
(124 entries)



Package Multi-Serial
(11 entries)



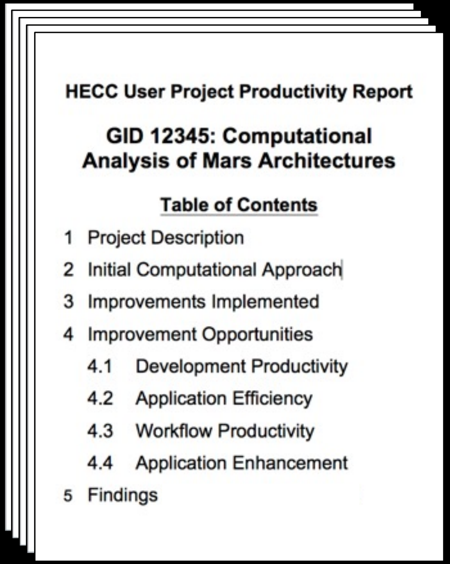
- ➢ Most applications (91%) are parallel.
- ➢ For serial applications, packaging multi-serial is mostly done with Job Array or GNU Parallel.

Programming Challenges

- Complex target hardware architectures/environments
  - CPUs with increasing number of cores, deep memory hierarchies; accelerators; vector engines, GPUs, FPGAs, heterogeneous environments, complex I/O infrastructure
- Multitude of programming models and environments
  - Programming languages and libraries: C/C++, Fortran, OpenMP, MPI
  - Multiple levels of parallelism
  - Offload for accelerators: OpenACC, OpenMP target, NVIDIA CUDA, AMD HIP, Intel oneAPI, SYCL
  - Scripting languages and frameworks: Python, Julia, R, Kokkos, Raja
  - Domain-specific application frameworks and libraries
- Users want both code and performance portability
- Large legacy code-bases – approaches?
  - Optimize existing code with some restructuring of code and data structures
  - Major rewrite to match architectures
  - Use different/more appropriate algorithms
- Lack of budget and expert labor resources

Approaches to overcome challenges

- Develop mini-apps for benchmarking
- Conduct hackathons partnering small teams of developers with expert mentor to develop expertise on emerging systems
- Form joint team of HPC experts with project scientists
  - For heavily utilized codes
  - Analyze code/workflow
  - Identify challenges/opportunities
  - Develop a strategy
  - Implement the strategy


HECC User Project Productivity Report

GID 12345: Computational Analysis of Mars Architectures

Table of Contents

Questions?

piyush.Mehrotra@nasa.gov

https://nas.nasa.gov